#2

Jc978 U.S. PTO
09/916690
07/30/01

Docket No.: P-241 PATENT

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re Application of	:	
Nam-Jin KIM and Hyung-Soo PARK	:	
Serial No.: To Be Assigned	:	Group Art Unit: To Be Assigned
Confirm. No.: To Be Assigned	:	Examiner: To Be Assigned
Filed: July 30, 2001	:	

For: APPARATUS AND METHOD FOR DATABASE SYNCHRONIZATION IN A DUPLEX SYSTEM

TRANSMITTAL OF CERTIFIED PRIORITY DOCUMENT

Assistant Commissioner of Patents
Washington, D. C. 20231

Sir:

At the time the above application was filed, priority was claimed based on the following application:

Korean Patent Application No. 43983/2000, filed July 29, 2000.

A copy of each priority application listed above is enclosed.

Respectfully submitted,
FLESHNER & KIM, LLP

Daniel Y.J. Kim
Registration No. 36,186
David W. Ward
Registration No. 45,198

P. O. Box 221200
Chantilly, Virginia 20153-1200
703 502-9440
Date: July 30, 2001

Tc970 U.S. PTO
09/916690
07/30/01

대 한 민 국 특 허 청
KOREAN INTELLECTUAL PROPERTY OFFICE

별첨 사본은 아래 출원의 원본과 동일함을 증명함.

This is to certify that the following application annexed hereto is a true copy from the records of the Korean Intellectual Property Office.

출 원 번 호 : Application Number	특허출원 2000년 제 43983 호 PATENT-2000-0043983
출 원 년 월 일 : Date of Application	2000년 07월 29일 JUL 29, 2000
출 원 인 : Applicant(s)	엘지정보통신주식회사 LG INFORMATION & COMMUNICATIONS LTD.

2001 년 07 월 13 일

특 허 청

COMMISSIONER

【서류명】	특허출원서
【권리구분】	특허
【수신처】	특허청장
【제출일자】	2000.07.29
【발명의 명칭】	이중화 시스템 환경에서 데이터 동기화를 위한 트랜잭션 관리 방법
【발명의 영문명칭】	Transaction Management Method For Data Synchronous In Dual System Environment
【출원인】	
【명칭】	엘지정보통신 주식회사
【출원인코드】	1-1998-000286-1
【대리인】	
【성명】	김영철
【대리인코드】	9-1998-000040-3
【포괄위임등록번호】	1999-010680-1
【발명자】	
【성명의 국문표기】	김남진
【성명의 영문표기】	KIM,Nam Jin
【주민등록번호】	720914-1031512
【우편번호】	135-284
【주소】	서울특별시 강남구 대치4동 동아아파트 가동 801호
【국적】	KR
【취지】	특허법 제42조의 규정에 의하여 위와 같이 출원합니다. 대리인 김영철 (인)
【수수료】	
【기본출원료】	20 면 29,000 원
【가산출원료】	5 면 5,000 원
【우선권주장료】	0 건 0 원
【심사청구료】	0 항 0 원
【합계】	34,000 원
【첨부서류】	1. 요약서·명세서(도면)_1통

【요약서】

【요약】

본 발명은 지능망이나 엔지니어링 워크스테이션 등과 같은 이중화 시스템 환경에서 트랜잭션 리스트를 기록하는 CL(Change List) 테이블을 사용하여 트랜잭션을 처리해서 데이터 동기화를 유지하도록 한 이중화 시스템 환경에서 데이터 동기화를 위한 트랜잭션 관리 방법에 관한 것이다.

종래의 이중화 시스템 환경에서 데이터 동기화를 위한 트랜잭션 관리 기법은 로그 레코드나 데이터 큐를 사용함에 따라 메모리 용량의 낭비를 피할 수 없으며, 상대적으로 느린 디스크 장치의 접근 시간과 많은 데이터 전송 시간 때문에 디스크 장치에 대해 신속한 데이터 접근을 보장하는 시스템의 성능 저하를 초래하게 되는 문제점이 있었다.

본 발명은 이중화 시스템 환경에서 트랜잭션 리스트를 기록하는 데이터베이스의 CL 테이블을 참고하여, 이전에 실패한 트랜잭션을 리모트 시스템에 반영함으로써, 이중화 시스템 환경에서 실시간으로 데이터 동기화를 유지할 수 있게 된다.

또한, 본 발명은 데이터베이스의 CL 테이블을 참고하여 데이터 동기화를 유지함으로써, 트랜잭션 관리와 관련하여 디스크 장치에 대한 접근 시간과 데이터 전송 시간을 줄일 수 있기 때문에 시스템 성능 저하를 방지할 수 있게 된다.

【대표도】

도 2

【명세서】

【발명의 명칭】

이중화 시스템 환경에서 데이터 동기화를 위한 트랜잭션 관리 방법{Transaction Management Method For Data Synchronous In Dual System Environment}

【도면의 간단한 설명】

도 1은 본 발명에 따른 이중화 시스템 환경에서 데이터 동기화를 위한 트랜잭션 관리 구조를 개략적으로 도시한 도면.

도 2는 본 발명에 따른 이중화 시스템 환경에서 데이터 동기화를 위한 트랜잭션 관리 절차를 도시한 순서도.

도 3은 도 2에 있어, 현재 발생한 트랜잭션의 오퍼레이션이 변경 오퍼레이션인 경우의 트랜잭션 관리 절차를 도시한 순서도.

도 4는 도 2에 있어, 현재 발생한 트랜잭션의 오퍼레이션이 삽입 오퍼레이션인 경우의 트랜잭션 관리 절차를 도시한 순서도.

도 5는 도 2에 있어, 현재 발생한 트랜잭션의 오퍼레이션이 삭제 오퍼레이션인 경우의 트랜잭션 관리 절차를 도시한 순서도.

* 도면의 주요 부분에 대한 부호의 설명 *

11, 21 : 데이터베이스 12, 22 : 질의처리 인터페이스 라이브러리

13, 23 : 응용 프로세스 14, 24 : 메시지 송신 프로세스

15, 25 : 메시지 수신 프로세스

【발명의 상세한 설명】

【발명의 목적】

【발명이 속하는 기술분야 및 그 분야의 종래기술】

<10> 본 발명은 이중화 시스템 환경에서 데이터 동기화를 위한 트랜잭션 관리 방법에 관한 것으로, 특히 지능망이나 엔지니어링 워크스테이션 등과 같은 이중화 시스템 환경에서 트랜잭션 리스트를 기록하는 CL(Change List) 테이블을 사용하여 트랜잭션을 처리해서 데이터 동기화를 유지하도록 한 이중화 시스템 환경에서 데이터 동기화를 위한 트랜잭션 관리 방법에 관한 것이다.

<11> 최근의 지능형 정보 서비스를 제공하는 지능망(Intelligent Network)이나 엔지니어링 워크스테이션(EngineeringWorkstation) 등과 같이 대량의 데이터를 빈번하게 검색하거나 변경하는 시스템에서는 보존하여야 할 데이터를 데이터베이스화해서 대용량의 디스크 장치에 저장시켜 보존한다.

<12> 하지만, 대용량 데이터베이스를 저장하는 디스크 장치는, 시스템 부하량의 변동이나 예기치 못한 기계적인 사고 등에 의하여 장애를 일으킬 수 있고, 이러한 기억 장치의 장애로 인하여 엄청난 양의 데이터가 일순간에 유실되는 문제점이 있었다.

<13> 그리하여, 상기와 같은 문제점을 해결하기 위한 종래의 방식으로서, 시스템과 데이터베이스를 이중화하여, 하나의 시스템(로컬 시스템)에 장애가 발생한 경우에 나머지 시스템(리모트 시스템)으로 대체하는 방식을 사용하였는데, 이 경우 각각의 데이터베이스를 관리하는 이중화 시스템 환경에서 데이터의 동기화를 보장하기 위해 사용하는 종래의

트랜잭션(transaction) 관리 기법들은 주로 로그(log) 레코드에 현재 처리한 트랜잭션에 관련된 데이터를 기록하여 이중화된 리모트 시스템으로 전송하는 방법을 사용하거나, 로컬 시스템과 리모트 시스템이 서로 동일한 용량의 데이터 큐를 운용하여 해당 로컬 시스템에서 트랜잭션 관련 데이터를 저장하고 이를 리모트 시스템의 데이터 큐에 반영하는 방법을 사용했다.

<14> 전술한 바와 같이, 종래의 이중화 시스템 환경에서 데이터 동기화를 위한 트랜잭션 관리 기법은 로그 레코드나 데이터 큐를 사용함에 따라 메모리 용량의 낭비를 피할 수 없으며, 상대적으로 느린 디스크 장치의 접근 시간과 많은 데이터 전송 시간 때문에 디스크 장치에 대해 신속한 데이터 접근을 보장하는 시스템의 성능 저하를 초래하게 되는 문제점이 있었다.

【발명이 이루고자 하는 기술적 과제】

<15> 본 발명은 전술한 바와 같은 문제점을 해결하기 위한 것으로 그 목적은, 이중화 시스템 환경에서 트랜잭션 리스트를 기록하는 데이터베이스의 CL 테이블을 참고하여, 이전에 실패한 트랜잭션을 리모트 시스템에 반영함으로써, 이중화 시스템 환경에서 실시간으로 데이터 동기화를 유지할 수 있도록 하는데 있다.

<16> 본 발명의 다른 목적은, 데이터베이스의 CL 테이블을 참고하여 데이터 동기화를 유지함으로써, 트랜잭션 관리와 관련된 디스크 장치에 대한 접근 시간과 데이터 전송 시간을 줄여 시스템 성능 저하를 방지할 수 있도록 하는데 있다.

【발명의 구성 및 작용】

<17> 상술한 바와 같은 목적을 해결하기 위한 본 발명의 특징은, 로컬 시스템에서 데이터베이스의 내용을 변경하는 트랜잭션을 처리하고, 해당되는 트랜잭션 리스트를 상기 데이터베이스의 CL 테이블에 기록한 후, 상기 트랜잭션 결과를 리모트 시스템에 반영하여 데이터 동기화를 유지하는 이중화 시스템 환경에 있어서, 상기 로컬 시스템에 트랜잭션이 발생하는 경우 해당되는 트랜잭션 처리를 수행한 후, 현재 발생한 트랜잭션이 데이터베이스의 내용을 변경하는 오퍼레이션인지를 확인하는 과정과; 현재 발생한 트랜잭션이 데이터베이스의 내용을 변경하는 오퍼레이션인 경우 상기 데이터베이스의 CL 테이블에 이전에 기록된 트랜잭션 리스트가 존재하는지를 확인하는 과정과; 상기 CL 테이블에 이전에 기록된 트랜잭션 리스트가 존재하는 경우 상기 CL 테이블에 존재하는 트랜잭션 리스트의 오퍼레이션 종류에 따라 리모트 시스템에 해당되는 트랜잭션 결과를 반영하여 데이터 동기화를 유지하는 과정을 포함하는 이중화 시스템 환경에서 데이터 동기화를 위한 트랜잭션 관리 방법을 제공하는데 있다.

<18> 이하, 본 발명에 따른 실시예를 첨부한 도면을 참조하여 상세하게 설명하면 다음과 같다.

<19> 본 발명에 따른 이중화 시스템 환경에서 데이터 동기화를 위한 트랜잭션 관리 구조는 첨부한 도면 도 1에 도시한 바와 같이, 동일한 구조의 데이터베이스(11, 21)를 갖는 이중화된 시스템(A측 시스템, B측 시스템)이 TCP/IP(Transmission Control Protocol/Internet Protocol) 소켓을 통해 연동하며, 각각의 시스템(10, 20)은 데이터베이스(11, 21)와, 두 개의 질의처리 인터페이스 라이브러리(Query Interface Library ;

12, 22)와, 응용 프로세스(13, 23)와, 메시지 송신 프로세스(14, 24) 및 메시지 수신 프로세스(15, 25)를 포함한다.

<20> 해당 데이터베이스(11, 21)는 디스크 장치(도면에 도시되어 있지 않음)에 존재하며, 데이터 테이블 이외에 트랜잭션 처리후의 데이터 동기화를 위해 트랜잭션 리스트 즉, 트랙잭션의 종류를 나타내는 오퍼레이션 코드와 트랜잭션이 발생한 테이블의 인덱스와 키값을 기록할 CL(Change List) 테이블을 포함하는데, 해당 트랜잭션의 종류에는 삽입(Insert), 삭제(Delete), 변경(Update) 오퍼레이션이 있으며, 해당 CL 테이블은 트랙잭션이 최초로 발생한 시스템(이하, '로컬 시스템'이라 칭하고, 설명의 편의를 위해 A측 시스템(10)을 로컬 시스템으로 가정함)에서 트랜잭션 리스트를 기록한 후, 동일한 트랜잭션 결과를 상대측 시스템(이하, '리모트 시스템'이라 칭하고, 설명의 편의를 위해 B측 시스템(20)을 리모트 시스템으로 가정함)에 성공적으로 반영한 후에 해당 CL 테이블의 트랜잭션 리스트를 삭제하되, 리모트 시스템으로의 트랜잭션 결과 반영에 실패하는 경우에는 이후에 발생되는 트랜잭션 처리시의 데이터 동기화를 위해 CL 테이블의 트랜잭션 리스트를 유지한다.

<21> 해당 질의처리 인터페이스 라이브러리(12, 22)는 응용 프로세스(13, 23)에 의해 데이터 동기화가 필요한 트랜잭션인 삽입, 삭제, 변경 오퍼레이션이 발생하는

경우 데이터베이스(11, 21)의 CL 테이블에 트랜잭션 리스트를 기록하여 소정의 트랜잭션을 수행하고, 트랜잭션을 성공적으로 수행한 경우 리모트 시스템 측에 동일한 트랜잭션 결과를 반영하기 위해 메시지 송신 프로세스(14, 24)로 트랜잭션 요구 메시지를 전달한 후, 리모트 시스템 측의 트랜잭션 반영이 성공했음을 나타내는 트랜잭션 완료 메시지를 통보받게 되면 해당 데이터베이스(11, 21)의 CL 테이블에 기록한 트랜잭션 리스트를 삭제한다.

<22> 해당 응용 프로세스(13, 23)는 질의처리 인터페이스 라이브러리(12, 22)를 통해 데이터베이스(11, 21)에 접근하여 트랜잭션 처리를 위한 질의를 수행한다.

<23> 해당 메시지 송신 프로세스(14, 24)는 질의처리 인터페이스 라이브러리(12, 22)로부터 트랜잭션 요구 메시지가 전달되는 경우 TCP/IP 소켓을 생성하여 리모트 시스템 측의 메시지 수신 프로세스(15, 25)와 연결을 성립한 후, 해당 리모트 시스템 측에 동일한 트랜잭션 결과를 반영하기 위해 질의처리 인터페이스 라이브러리(12, 22)로부터 전달된 트랜잭션 요구 메시지를 송신한다.

<24> 해당 메시지 수신 프로세스(15, 25)는 로컬 시스템 측에서 생성한 TCP/IP 소켓을 통해 트랜잭션 요구 메시지를 수신받아 질의처리 인터페이스 라이브러리(12, 22) 측으로 전달하거나, 리모트 시스템 측으로부터 트랜잭션 반영이 성공했음을 나타내는 트랜잭션 결과 메시지를 수신받아 질의처리 인터페이스 라이브러리(12, 22) 측으로 전달한다.

<25> 이와 같이 구성된 본 발명에 따른 이중화 시스템 환경에서 데이터 동기화를 위한 트랜잭션 관리 방법을 동기화 동작 절차와 동기화 트랜잭션 동작 절차로 구분하여 설명하면 다음과 같다.

<26> 먼저, 동기화 동작 절차는 데이터베이스(11, 21)의 내용을 삽입/삭제/변경하는 오퍼레이션을 나타내는 트랜잭션이 이중화된 로컬 시스템(10)에서 발생하는 경우에 해당 트랜잭션 처리를 위해 리모트 시스템(20)과 어떻게 동작하는가에 관한 절차를 의미하는 것으로, 우선은 소정의 트랜잭션이 발생한 로컬 시스템(10)에서 현재 발생한 트랜잭션을 처리한 후, 동기화 트랜잭션 처리 절차에 따라 이중화된 리모트 시스템(20) 측으로 동일한 트랜잭션 결과를 반영하기 위해 트랜잭션 요구 메시지를 송신하게 된다.

<27> 즉, 해당 로컬 시스템(10)의 응용 프로세스(13)가 소정의 메시지를 수신받게 되면, 질의처리 인터페이스 라이브러리(12)를 통해 데이터베이스(12)에 접근한 후, 해당되는 트랜잭션 처리를 수행하게 된다.

<28> 이때, 해당 질의처리 인터페이스 라이브러리(12)는 데이터 동기화가 필요한 트랜잭션 처리가 정상적으로 수행되면 데이터베이스(11)의 CL 테이블에 트랜잭션 리스트 즉, 오퍼레이션 코드와 테이블의 인덱스 및 키값을 기록하게 된다.

<29> 이후, 해당 질의처리 인터페이스 라이브러리(12)는 데이터베이스(11)의 CL 테이블에 트랜잭션 리스트가 정상적으로 기록되는지를 확인하여, 해당 트랜잭션 리스트가 정상적으로 기록되는 경우 이중화된 리모트 시스템(20)에 동일한 트랜잭션 결과를 반영하기 위해 메시지 송신 프로세스(14)를 통해 리모트 시스템(20) 측에 트랜잭션 요구 메시지를 송신하게 된다.

<30> 그런데, 해당 데이터베이스(11)의 CL 테이블에 트랜잭션 리스트가 정상적으로 기록되지 않은 경우 즉, CL 테이블에 트랜잭션 리스트의 기록을 실패하는 경우에는 해당 메시지에 대해 실패 처리를 한 후, 해당 트랜잭션에 대한 작업을 회복하되, 해당 회복 작업이 실패하게 되면 로그 파일에 기록하게 된다.

<31> 한편, 해당 로컬 시스템(20)의 질의처리 인터페이스 라이브러리(22)는 메시지 수신 프로세스(25)를 감시하여, 트랜잭션 요구 메시지를 송신한 리모트 시스템(20)으로부터 트랜잭션 반영이 성공적했음을 나타내는 트랜잭션 결과 메시지가 수신되는지를 확인하여, 해당 트랜잭션 결과 메시지가 수신되는 경우에는 데이터베이스(11)의 CL 테이블에 기록한 트랜잭션 리스트를 삭제하게 되고, 해당 트랜잭션 결과 메시지가 수신되지 않는 경우 즉, 리모트 시스템(20) 측에서 트랜잭션 반영에 실패한 경우에는 다음에 발생하는 트랜잭션을 통해 데이터 동기를 회복하기 위해서 해당 데이터베이스(11)의 CL 테이블에 기록한 트랜잭션 리스트를 유지하게 된다.

<32> 다음으로, 동기화 트랜잭션 동작 절차 즉, 트랜잭션을 처리하는 이중화된 시스템(10, 20)간에 데이터 동기화를 위해서 트랜잭션을 어떻게 처리하는가에 관한 트랜잭션 관리 절차를 첨부한 도면 도 2를 참조하여 설명하면 다음과 같다.

<33> 먼저, 후술하고자 하는 동기화 트랜잭션 동작은 로컬 시스템(10)에 포함된 데이터베이스(11)의 CL 테이블에 이전에 기록된 트랜잭션 리스트가 존재하는 경우에 수행하게 됨에 따라, 로컬 시스템(10)과 리모트 시스템(20) 모두에서 트랜잭션을 성공적으로 처리하여 CL 테이블에 기록했던 트랜잭션 리스트를 삭제한 경우나 로컬 시스템(10)에서 트랜잭션 처리를 실패하여 CL 테이블에 트랜잭션 리스트를 기록하지 않은 경우에 대해서는 그 설명을 생략하기로 한다.

<34> 한편, 해당 동기화 트랜잭션 동작은 이중화 시스템 환경에서 로컬 시스템(10)에 트랜잭션이 발생하는 경우 해당 트랜잭션이 발생한 로컬 시스템(10)의 응용 프로세스(13)에서 질의처리 인터페이스 라이브러리(12)를 통해 데이터베이스(11)의 데이터 테이블에 해당되는 트랜잭션 처리를 수행하게 되면(스텝 S21, S22), 해당 질의처리 인터페이스 라

이브러리(12)에서는 데이터베이스(11)의 CL 테이블에 트랜잭션 리스트 즉, 오퍼레이션 코드와 테이블의 인덱스 및 키값을 기록한 후(스텝 S23), 현재 발생한 트랜잭션이 데이터베이스(11)의 내용을 변경하는 삽입/삭제/변경 오퍼레이션인지를 확인하게 된다(스텝 S24).

<35> 이때, 해당 데이터베이스(11)의 내용을 변경하는 오퍼레이션인 경우 해당 질의처리 인터페이스 라이브러리(12)는 데이터베이스(11)의 CL 테이블에 이전에 기록된 트랜잭션 리스트가 존재하는지를 확인하여(스텝 S25), 해당 CL 테이블에 이전에 기록된 트랜잭션 리스트가 존재하지 않는 경우에는 메시지 송신 프로세스(14)를 통해 이중화된 리모트 시스템(20) 측으로 트랜잭션 요구 메시지를 송신하여 동일한 트랜잭션 결과를 반영하게 된다(스텝 S26).

<36> 그런데, 스텝 S25에서 해당 데이터베이스(11)의 CL 테이블에 이전에 기록된 트랜잭션 리스트가 존재하는 것으로 확인되는 경우 즉, 이전에 발생한 트랜잭션에 대해 로컬 시스템(10)에서는 트랜잭션을 정상적으로 처리했지만 리모트 시스템(20)에서는 트랜잭션 반영에 실패함에 따라 데이터베이스(11)의 CL 테이블에 기록한 트랜잭션 리스트를 삭제하지 않고 유지시켜 놓은 경우 해당 로컬 시스템(10)의 질의처리 인터페이스 라이브러리(12)는 CL 테이블에 존재하는 트랜잭션 리스트의 오퍼레이션 종류를 확인한 후(스텝 S27), 해당 CL 테이블의 오퍼레이션 종류에 따라 리모트 시스템(20)에 해당되는 트랜잭션 결과를 반영함으로써(스텝 S28), 해당 로컬 시스템(10)과 리모트 시스템(20)간의 데이터 동기화를 유지하게 된다.

<37> 나아가, 상술한 동기화 트랜잭션 동작 절차를 현재 발생한 트랜잭션의 오퍼레이션 종류에 따라 보다 상세히 설명하면 다음과 같다.

<38> 첫째로, 현재 발생한 트랜잭션의 오퍼레이션이 변경 오퍼레이션인 경우 첨부한 도면 도 3을 참조하여 설명하면, 해당 로컬 시스템(10)의 응용 프로세스(13)는 데이터베이스(11)의 데이터 테이블에 변경 오퍼레이션을 처리한 후(스텝 S31), 키값을 이용하여 데이터베이스(11)의 CL 테이블에 이전에 기록된 트랜잭션 리스트가 존재하는지를 확인하게 된다(스텝 S32).

<39> 이때, 해당 CL 테이블에 이전에 기록된 트랜잭션 리스트가 존재하는 것으로 확인되면, 해당 키값을 갖는 튜플에 대해서 이중화된 로컬 시스템(10)과 리모트 시스템(20)간에 데이터 동기화가 실패한 트랜잭션이 발생했던 것이므로, 해당 CL 테이블의 오퍼레이션 종류에 따라 리모트 시스템(20)에 해당되는 트랜잭션 결과를 반영하게 되는데, 이를 위해 해당 로컬 시스템(10)의 질의처리 인터페이스 라이브러리(12)는 CL 테이블에 존재하는 트랜잭션 리스트의 오퍼레이션 종류를 확인하게 된다(스텝 S33).

<40> 만약, 해당 CL 테이블의 오퍼레이션 종류를 확인한 결과 '변경' 오퍼레이션이라면 이전에 발생한 변경 트랜잭션이 리모트 시스템(20)에 반영되지 않았으므로, 현재 발생한 변경 트랜잭션을 리모트 시스템(20)에 반영한 후(스텝 S34), 해당 데이터베이스(11)의 CL 테이블에 존재하는 트랜잭션 리스트를 삭제하게 된다(스텝 S35).

<41> 그런데, 해당 CL 테이블의 오퍼레이션 종류를 확인한 결과 '삽입' 오퍼레이션이라면 이전에 발생한 삽입 트랜잭션이 리모트 시스템(20)에 반영되지 않았으므로, 현재 발생한 변경 트랜잭션을 삽입 트랜잭션으로 리모트 시스템(20)에 반영한 후(스텝 S36), 해당 데이터베이스(11)의 CL 테이블에 존재하는 트랜잭션 리스트를 삭제하게 된다(스텝 S35).

<42> 참고로, 현재 발생한 트랜잭션의 오퍼레이션이 변경 오퍼레이션인 경우에는 데이터

베이스(11)의 CL 테이블에 '삭제' 오퍼레이션이 존재할 수 없는데, 이는 로컬 시스템(10)에서 이전에 삭제 오퍼레이션을 수행하게 되면, 삭제된 튜플에 대해 다음에 변경 트랜잭션이 발생할 수 없기 때문이다.

<43> 둘째로, 현재 발생한 트랜잭션의 오퍼레이션이 삽입 오퍼레이션인 경우 첨부한 도면 도 4를 참조하여 설명하면, 해당 로컬 시스템(10)의 응용 프로세스(13)는 데이터베이스(11)의 데이터 테이블에 삽입 오퍼레이션을 처리한 후(스텝 S41), 키값을 이용하여 데이터베이스(11)의 CL 테이블에 이전에 기록된 트랜잭션 리스트가 존재하는지를 확인하게 된다(스텝 S42).

<44> 이때, 해당 CL 테이블에 이전에 기록된 트랜잭션 리스트가 존재하는 것으로 확인되면, 해당 키값을 갖는 튜플에 대해서 이중화된 로컬 시스템(10)과 리모트 시스템(20)간에 데이터 동기화가 실패한 트랜잭션이 발생했던 것이므로, 해당 CL 테이블의 오퍼레이션 종류에 따라 리모트 시스템(20)에 해당되는 트랜잭션 결과를 반영하게 되는데, 이를 위해 해당 로컬 시스템(10)의 질의처리 인터페이스 라이브러리(12)는 CL 테이블에 존재하는 트랜잭션 리스트의 오퍼레이션 종류를 확인하게 된다(스텝 S43).

<45> 만약, 해당 CL 테이블의 오퍼레이션 종류를 확인한 결과 '삽입' 오퍼레이션이라면 이전에 발생한 삽입 트랜잭션이 리모트 시스템(20)에 반영되지 않았으므로, 현재 발생한 삽입 트랜잭션을 리모트 시스템(20)에 반영한 후(스텝 S44), 해당 데이터베이스(11)의 CL 테이블에 존재하는 트랜잭션 리스트를 삭제하게 된다(스텝 S45).

<46> 그런데, 해당 CL 테이블의 오퍼레이션 종류를 확인한 결과 '삭제' 오퍼레이션이라면 이전에 발생한 삭제 트랜잭션이 리모트 시스템(20)에 반영되지 않았으므로, 현재 발생한 삭제 트랜잭션을 변경 트랜잭션으로 리모트 시스템(20)에 반영한 후(스텝 S46), 해

당 데이터베이스(11)의 CL 테이블에 존재하는 트랜잭션 리스트를 삭제하게 된다(스텝 S45).

<47> 그리고, 해당 CL 테이블의 오퍼레이션 종류를 확인한 결과 '변경' 오퍼레이션이라면 상술한 '삭제' 오퍼레이션과 마찬가지로 이전에 발생한 삭제 트랜잭션과 변경 트랜잭션이 모두 리모트 시스템(20)에 반영되지 않았으므로, 현재 발생한 삽입 트랜잭션을 변경 트랜잭션으로 리모트 시스템(20)에 반영한 후(스텝 S46), 해당 데이터베이스(11)의 CL 테이블에 존재하는 트랜잭션 리스트를 삭제하게 된다(스텝 S45).

<48> 셋째로, 현재 발생한 트랜잭션의 오퍼레이션이 삭제 오퍼레이션인 경우 첨부한 도면 도 5를 참조하여 설명하면, 해당 로컬 시스템(10)의 응용 프로세스(13)는 데이터베이스(11)의 데이터 테이블에 삭제 오퍼레이션을 처리한 후(스텝 S51), 키값을 이용하여 데이터베이스(11)의 CL 테이블에 이전에 기록된 트랜잭션 리스트가 존재하는지를 확인하게 된다(스텝 S52).

<49> 이때, 해당 CL 테이블에 이전에 기록된 트랜잭션 리스트가 존재하는 것으로 확인되면, 해당 키값을 갖는 튜플에 대해서 이중화된 로컬 시스템(10)과 리모트 시스템(20)간에 데이터 동기화가 실패한 트랜잭션이 발생했던 것이므로, 해당 CL 테이블의 오퍼레이션 종류에 따라 리모트 시스템(20)에 해당되는 트랜잭션 결과를 반영하게 되는데, 이를 위해 해당 로컬 시스템(10)의 질의처리 인터페이스 라이브러리(12)는 CL 테이블에 존재하는 트랜잭션 리스트의 오퍼레이션 종류를 확인하게 된다(스텝 S53).

<50> 만약, 해당 CL 테이블의 오퍼레이션 종류를 확인한 결과 '변경' 오퍼레이션이라면 이전에 발생한 변경 트랜잭션이 리모트 시스템(20)에 반영되지 않았으므로, 현재 발생한 삭제 트랜잭션을 리모트 시스템(20)에 반영한 후(스텝 S54), 해당 데이터베이스(11)의

CL 테이블에 존재하는 트랜잭션 리스트를 삭제하게 된다(스텝 S55).

<51> 그런데, 해당 CL 테이블의 오퍼레이션 종류를 확인한 결과 '삽입' 오퍼레이션이라면 이전에 발생한 삽입 트랜잭션이 라모트 시스템(20)에 반영되지 않았으므로, 해당 리모트 시스템(20)에는 해당 튜플이 존재하지 않기 때문에 현재 발생한 삭제 트랜잭션은 리모트 시스템(20)에 반영하지 않고, 해당 데이터베이스(11)의 CL 테이블에 존재하는 트랜잭션 리스트를 삭제하게 된다(스텝 S55).

<52> 참고로, 현재 발생한 트랜잭션의 오퍼레이션이 삭제 오퍼레이션인 경우에는 데이터베이스(11)의 CL 테이블에 '삭제' 오퍼레이션이 존재할 수 없는데, 이는 로컬 시스템(10)에서 이전에 삭제 오퍼레이션을 수행하게 되면, 삭제된 튜플에 대해 다음에 삭제 트랜잭션이 발생할 수 없기 때문이다.

<53> 이로써, 본 발명에서는 대량의 데이터를 빈번하게 검색하거나 변경하면서도 두 시스템(10, 20)간의 데이터 동기화를 보장해야 하는 이중화 시스템 환경에서 데이터베이스(11)의 CL 테이블에 기록되어 있는 트랜잭션 리스트를 참고하여, 이전에 실패한 트랜잭션을 리모트 시스템(20)에 반영함으로써, 이중화 시스템 환경에서 실시간으로 데이터 동기를 유지할 수 있게 된다.

<54> 또한, 본 발명에 따른 실시예는 상술한 것으로 한정되지 않고, 본 발명과 관련하여 통상의 지식을 가진자에게 자명한 범위내에서 여러 가지의 대안, 수정 및 변경하여 실시할 수 있다.

【발명의 효과】

<55> 이상과 같이, 본 발명은 이중화 시스템 환경에서 트랜잭션 리스트를 기록하는 데이터베이스의 CL 테이블을 참고하여, 이전에 실패한 트랜잭션을 리모트 시스템에 반영함으로써, 이중화 시스템 환경에서 실시간으로 데이터 동기화를 유지할 수 있게 된다.

<56> 또한, 본 발명은 데이터베이스의 CL 테이블을 참고하여 데이터 동기화를 유지함으로써, 트랜잭션 관리와 관련하여 디스크 장치에 대한 접근 시간과 데이터 전송 시간을 줄일 수 있기 때문에 시스템 성능 저하를 방지할 수 있게 된다.

【특허청구범위】

【청구항 1】

로컬 시스템에서 데이터베이스의 내용을 변경하는 트랜잭션을 처리하고, 해당되는 트랜잭션 리스트를 상기 데이터베이스의 CL 테이블에 기록한 후, 상기 트랜잭션 결과를 리모트 시스템에 반영하여 데이터 동기화를 유지하는 이중화 시스템 환경에 있어서,

상기 로컬 시스템에 트랜잭션이 발생하는 경우 해당되는 트랜잭션 처리를 수행한 후, 현재 발생한 트랜잭션이 데이터베이스의 내용을 변경하는 오퍼레이션인지를 확인하는 과정과;

현재 발생한 트랜잭션이 데이터베이스의 내용을 변경하는 오퍼레이션인 경우 상기 데이터베이스의 CL 테이블에 이전에 기록된 트랜잭션 리스트가 존재하는지를 확인하는 과정과;

상기 CL 테이블에 이전에 기록된 트랜잭션 리스트가 존재하는 경우 상기 CL 테이블에 존재하는 트랜잭션 리스트의 오퍼레이션 종류에 따라 리모트 시스템에 해당되는 트랜잭션 결과를 반영하여 데이터 동기화를 유지하는 과정을 포함하는 것을 특징으로 하는 이중화 시스템 환경에서 데이터 동기화를 위한 트랜잭션 관리 방법.

【청구항 2】

제 1항에 있어서,

현재 발생한 트랜잭션의 오퍼레이션이 변경 오퍼레이션인 경우 상기 데이터베이스의 데이터 테이블에 변경 오퍼레이션을 처리하는 과정과;

키값을 이용하여 상기 데이터베이스의 CL 테이블에 이전에 기록된 트랜잭션 리스트가 존재하는지를 확인하는 과정과;

상기 CL 테이블에 이전에 기록된 트랜잭션 리스트가 존재하는 경우 상기 CL 테이블에 존재하는 트랜잭션 리스트의 오퍼레이션 종류를 확인하는 과정과;

상기 CL 테이블의 오퍼레이션 종류를 확인한 결과 '변경' 오퍼레이션이라면, 현재 발생한 변경 트랜잭션을 리모트 시스템에 반영한 후, 상기 데이터베이스의 CL 테이블에 존재하는 트랜잭션 리스트를 삭제하는 과정과;

상기 CL 테이블의 오퍼레이션 종류를 확인한 결과 '삽입' 오퍼레이션이라면, 현재 발생한 변경 트랜잭션을 삽입 트랜잭션으로 리모트 시스템에 반영한 후, 상기 데이터베이스의 CL 테이블에 존재하는 트랜잭션 리스트를 삭제하는 과정을 포함하는 것을 특징으로 하는 이중화 시스템 환경에서 데이터 동기화를 위한 트랜잭션 관리 방법.

【청구항 3】

제 1항에 있어서,

현재 발생한 트랜잭션의 오퍼레이션이 삽입 오퍼레이션인 경우 상기 데이터베이스의 데이터 테이블에 삽입 오퍼레이션을 처리하는 과정과;

키값을 이용하여 상기 데이터베이스의 CL 테이블에 이전에 기록된 트랜잭션 리스트가 존재하는지를 확인하는 과정과;

상기 CL 테이블에 이전에 기록된 트랜잭션 리스트가 존재하는 경우 상기 CL 테이블에 존재하는 트랜잭션 리스트의 오퍼레이션 종류를 확인하는 과정과;

상기 CL 테이블의 오퍼레이션 종류를 확인한 결과 '삽입' 오퍼레이션이라면, 현재 발생한 삽입 트랜잭션을 리모트 시스템에 반영한 후, 상기 데이터베이스의 CL 테이블에 존재하는 트랜잭션 리스트를 삭제하는 과정과;

상기 CL 테이블의 오퍼레이션 종류를 확인한 결과 '삭제' 오퍼레이션이라면, 현재 발생한 삭제 트랜잭션을 변경 트랜잭션으로 리모트 시스템에 반영한 후, 상기 데이터베이스의 CL 테이블에 존재하는 트랜잭션 리스트를 삭제하는 과정과;

상기 CL 테이블의 오퍼레이션 종류를 확인한 결과 '변경' 오퍼레이션이라면, 현재 발생한 삽입 트랜잭션을 변경 트랜잭션으로 리모트 시스템에 반영한 후, 상기 데이터베이스의 CL 테이블에 존재하는 트랜잭션 리스트를 삭제하는 과정을 포함하는 것을 특징으로 하는 이중화 시스템 환경에서 데이터 동기화를 위한 트랜잭션 관리 방법.

【청구항 4】

제 1항에 있어서,

현재 발생한 트랜잭션의 오퍼레이션이 삭제 오퍼레이션인 경우 상기 데이터베이스의 데이터 테이블에 삭제 오퍼레이션을 처리하는 과정과;

키값을 이용하여 상기 데이터베이스의 CL 테이블에 이전에 기록된 트랜잭션 리스트가 존재하는지를 확인하는 과정과;

상기 CL 테이블에 이전에 기록된 트랜잭션 리스트가 존재하는 경우 상기 CL 테이블에 존재하는 트랜잭션 리스트의 오퍼레이션 종류를 확인하는 과정과;

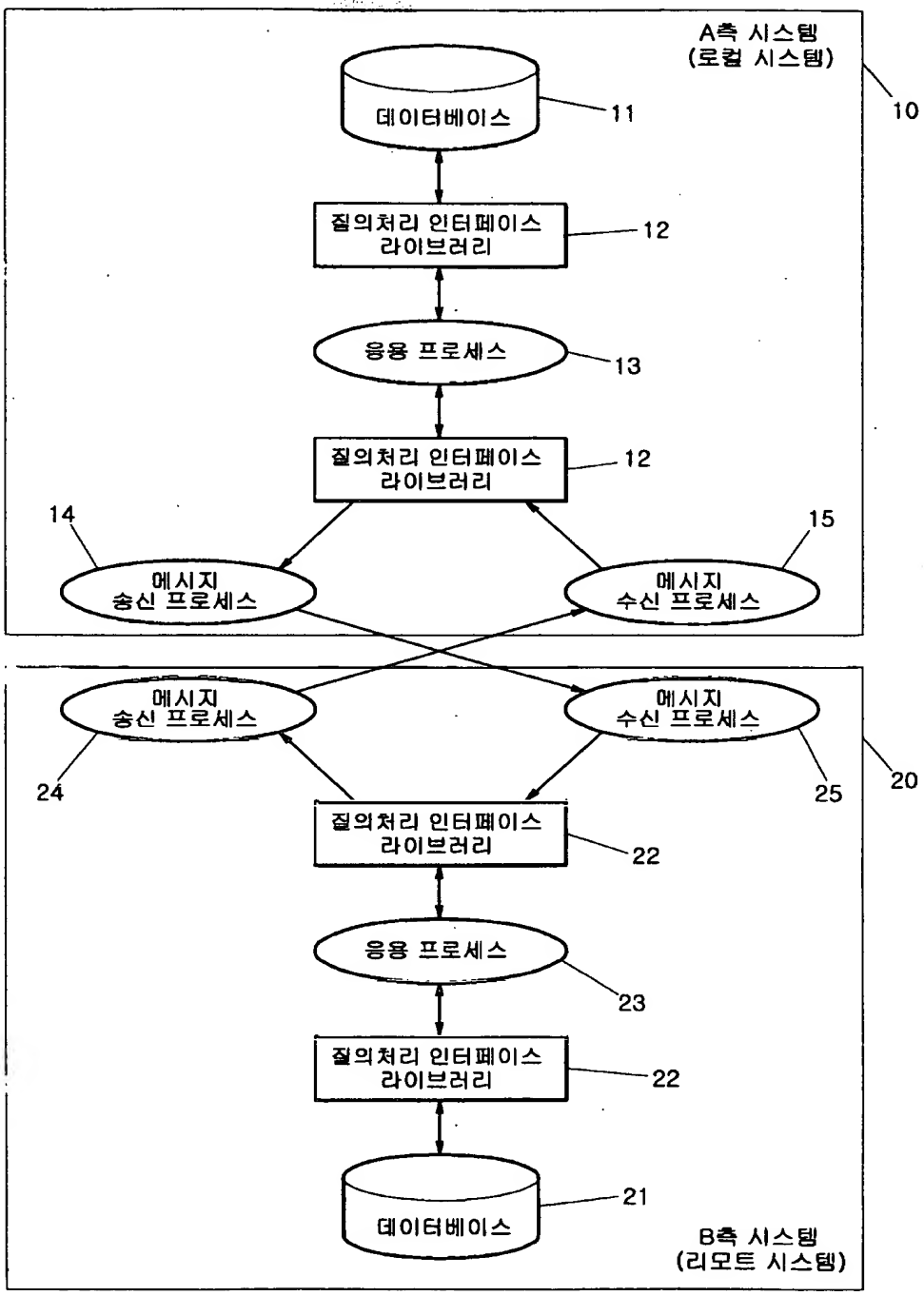
상기 CL 테이블의 오퍼레이션 종류를 확인한 결과 '변경' 오퍼레이션이라면, 현재

발생한 삭제 트랜잭션을 리모트 시스템에 반영한 후, 상기 데이터베이스의 CL 테이블에 존재하는 트랜잭션 리스트를 삭제하는 과정과;

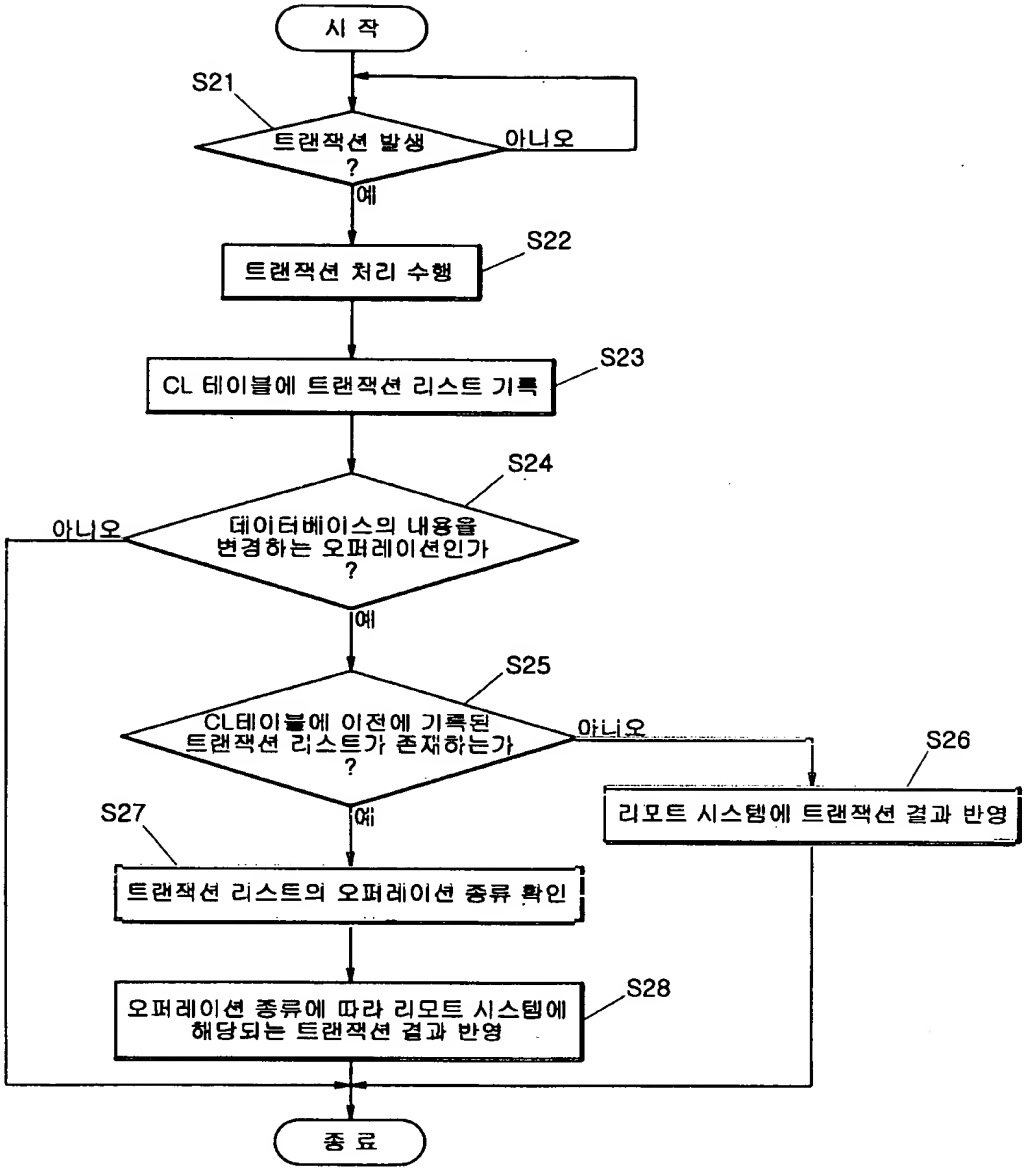
상기 CL 테이블의 오퍼레이션 종류를 확인한 결과 '삽입' 오퍼레이션이라면, 현재 발생한 삭제 트랜잭션을 리모트 시스템에 반영하지 않고, 상기 데이터베이스의 CL 테이블에 존재하는 트랜잭션 리스트를 삭제하는 과정을 포함하는 것을 특징으로 하는 이중화 시스템 환경에서 데이터 동기화를 위한 트랜잭션 관리 방법.

【도면】

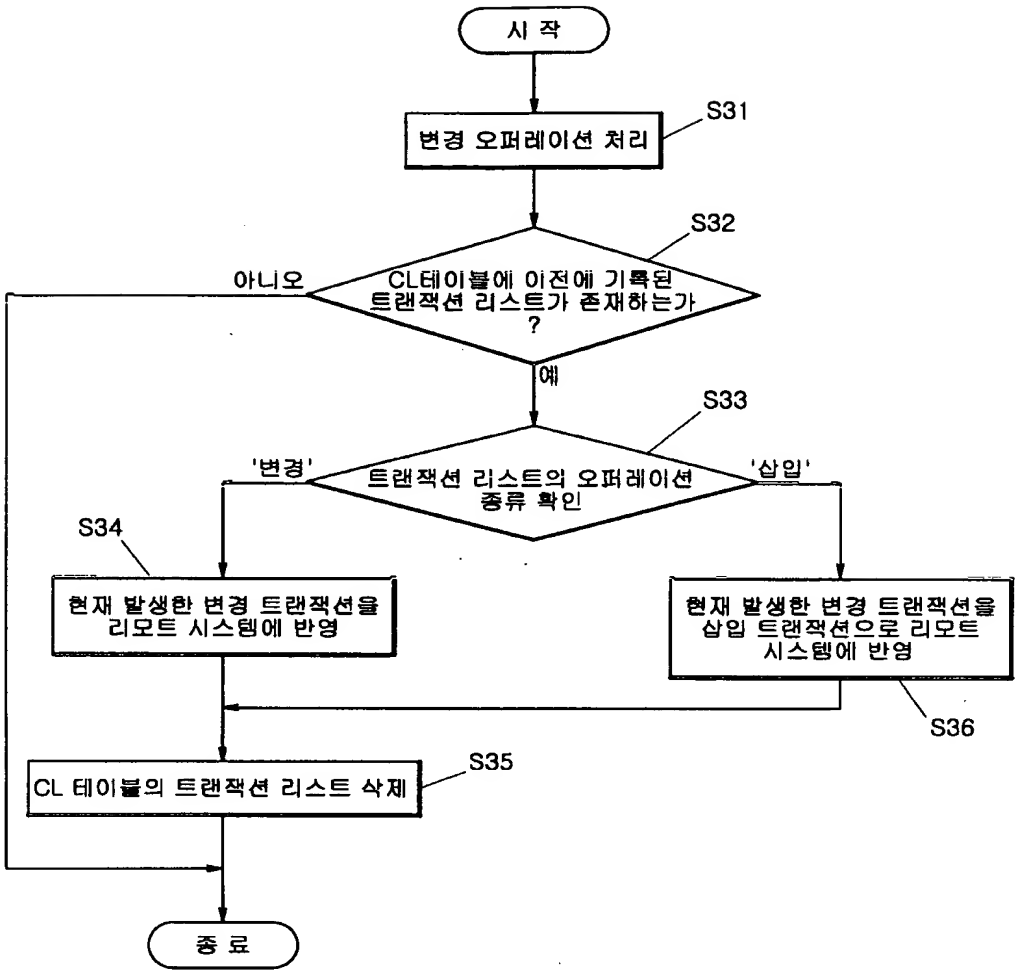
【도 1】


A측 시스템
(로컬 시스템)
10
데이터베이스
11
질의처리 인터페이스
라이브러리
12
응용 프로세스
13
질의처리 인터페이스
라이브러리
12
14
15
메시지
송신 프로세스
메시지
수신 프로세스
메시지
송신 프로세스
메시지
수신 프로세스
24
25
20
질의처리 인터페이스
라이브러리
22
응용 프로세스
23
질의처리 인터페이스
라이브러리
22
데이터베이스
21
B측 시스템
(리모트 시스템)

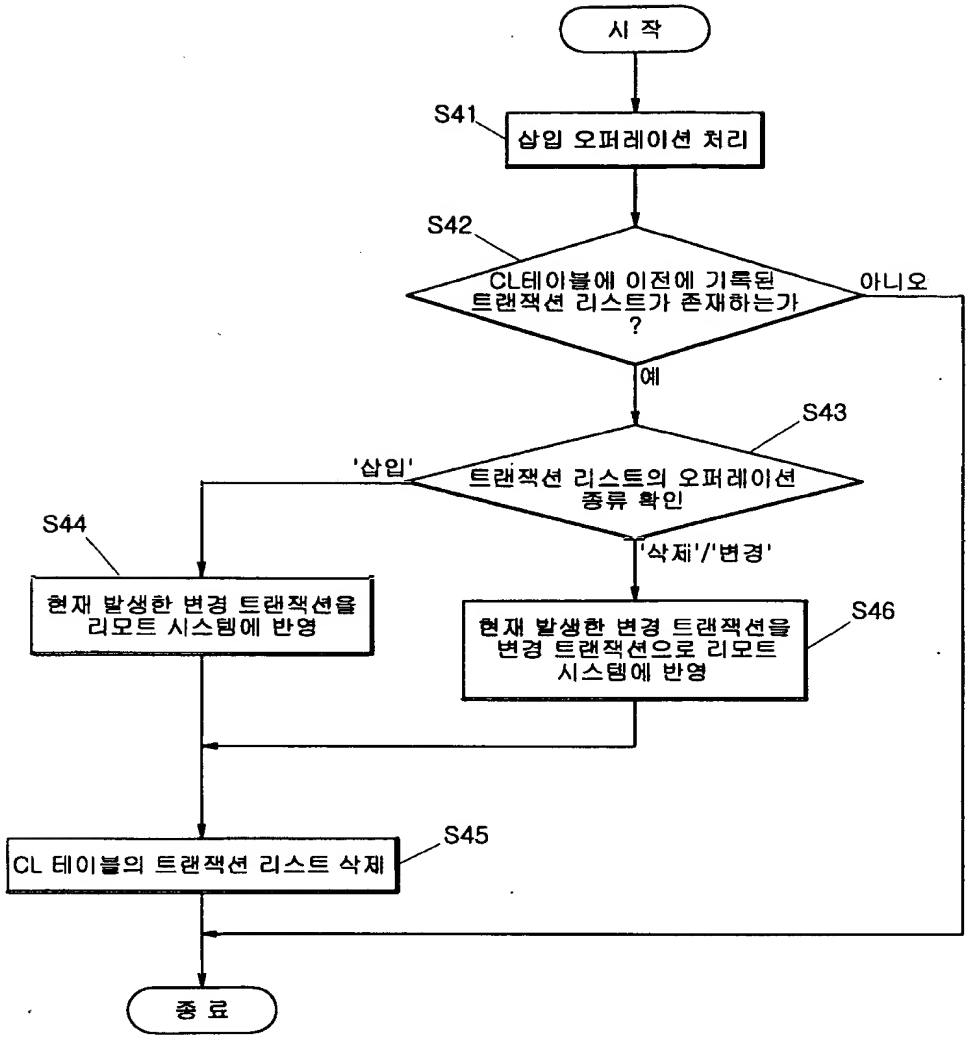
【도 2】


시 작
S21
트랜잭션 발생 ?
아니오
예
S22
트랜잭션 처리 수행
S23
CL 테이블에 트랜잭션 리스트 기록
S24
데이터베이스의 내용을 변경하는 오퍼레이션인가 ?
아니오
예
S25
CL테이블에 이전에 기록된 트랜잭션 리스트가 존재하는가 ?
아니오
S26
리모트 시스템에 트랜잭션 결과 반영
예
S27
트랜잭션 리스트의 오퍼레이션 종류 확인
S28
오퍼레이션 종류에 따라 리모트 시스템에 해당되는 트랜잭션 결과 반영
종 료

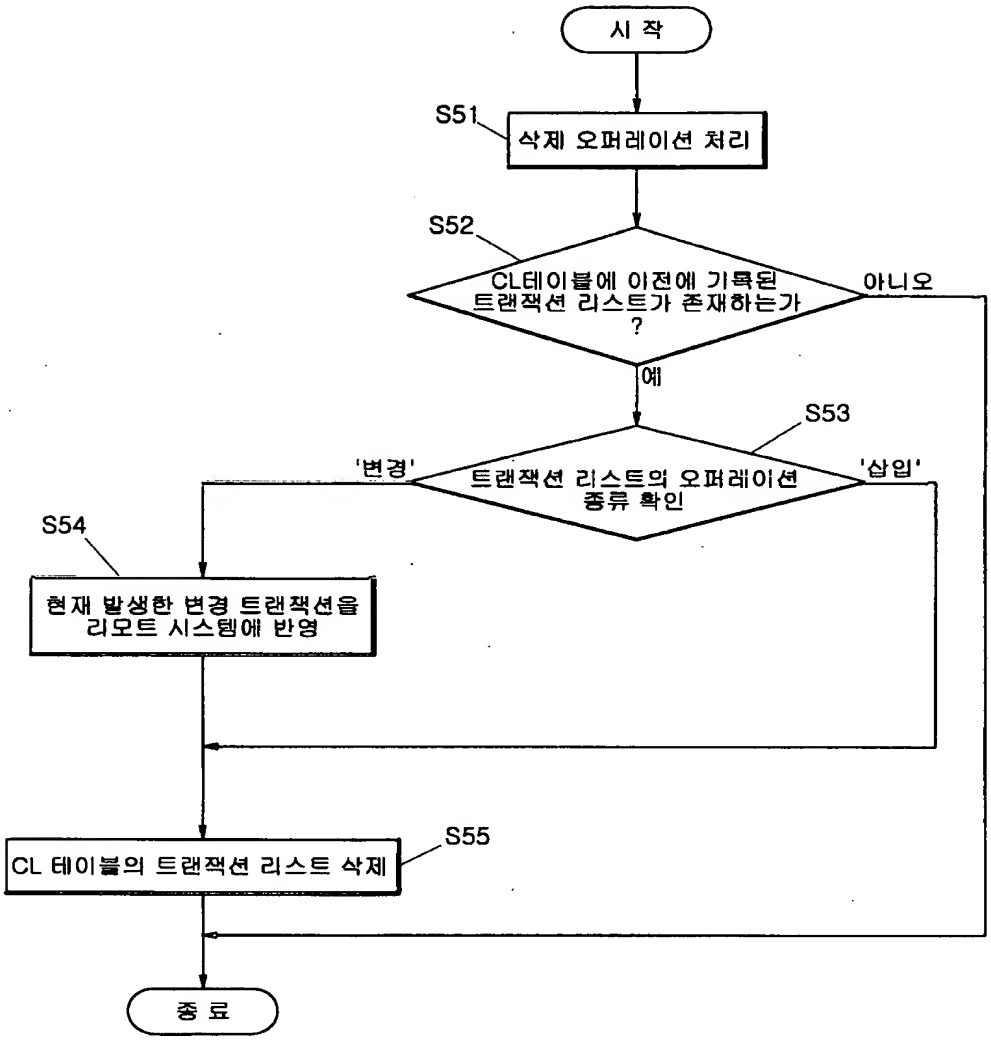
【도 3】


시 작
변경 오퍼레이션 처리
S31
S32
CL테이블에 이전에 기록된 트랜잭션 리스트가 존재하는가 ?
아니오
예
S33
'변경'
트랜잭션 리스트의 오퍼레이션 종류 확인
'삽입'
S34
현재 발생한 변경 트랜잭션을 리모트 시스템에 반영
현재 발생한 변경 트랜잭션을 삽입 트랜잭션으로 리모트 시스템에 반영
S36
CL 테이블의 트랜잭션 리스트 삭제
S35
종 료

【도 4】


시 작
S41
삽입 오퍼레이션 처리
S42
CL테이블에 이전에 기록된 트랜잭션 리스트가 존재하는가 ?
아니오
예
S43
'삽입'
트랜잭션 리스트의 오퍼레이션 종류 확인
'삭제'/'변경'
S44
현재 발생한 변경 트랜잭션을 리모트 시스템에 반영
현재 발생한 변경 트랜잭션을 변경 트랜잭션으로 리모트 시스템에 반영
S46
CL 테이블의 트랜잭션 리스트 삭제
S45
종 료

【도 5】


시 작
S51
삭제 오퍼레이션 처리
S52
CL테이블에 이전에 기록된 트랜잭션 리스트가 존재하는가 ?
아니오
예
S53
'변경'
트랜잭션 리스트의 오퍼레이션 종류 확인
'삽입'
S54
현재 발생한 변경 트랜잭션을 리모트 시스템에 반영
S55
CL 테이블의 트랜잭션 리스트 삭제
종 료